

El desarrollo de la comprensión lectora en las carreras de Informática

Sonia Rueda

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Abstract. El desarrollo de las **competencias comunicativas** involucra entre otras capacidades la **comprensión lectora**. En las carreras de Informática las materias iniciales demandan cierto nivel de desarrollo de esta capacidad pero también brindan la oportunidad de reforzarla. En estas asignaturas se utiliza lenguaje natural y distintos lenguajes artificiales. Este trabajo describe los diferentes modelos de textos y de lenguajes a través de los cuáles se fortalece la comprensión lectora, en cada una de las materias de Programación de las carreras ofrecidas por el Departamento de Ciencias e Ingeniería de la Computación de la UNS.

Keywords: Diseño curricular basado en competencias. Comprensión Lectora. Lenguaje natural y lenguaje artificial.

1 Introduction

La formación de un profesional en Informática requiere, entre otros aspectos, el desarrollo del razonamiento lógico, la argumentación, la organización de la información y la apropiación del lenguaje de la ciencia y la tecnología. Todas estas capacidades demandan a su vez de *competencias comunicativas* que permitan elaborar e interpretar *actos del habla*.

Cada nivel educativo presupone cierto nivel de desarrollo en las competencias comunicativas y asume la responsabilidad de reforzarlo. Sin embargo, en el ámbito universitario es poco frecuente que se establezca con precisión lo que se asume adquirido, lo que se espera desarrollar y las acciones y actividades concretas que se realizan para alcanzar este desarrollo.

La *comprensión lectora* es una capacidad básica fundamental para la competencia comunicativa de un alumno que comienza a cursar una carrera en el nivel superior. Es también imprescindible para un graduado de una carrera universitaria. Ahora bien, ¿qué nivel de comprensión lectora se espera de un ingresante? ¿cuál debería ser el nivel de desarrollo de esta capacidad en un graduado?

El objetivo de este trabajo es analizar y describir los diferentes *modelos de textos*, *elementos discursivos* y *lenguajes* con los cuales se fortalece la comprensión lectora en las materias de Programación de las carreras ofrecidas por el Departamento de Ciencias e Ingeniería de la Computación (DCIC) de la Universidad Nacional del Sur (UNS). Los programas de estas materias incluyen a los contenidos básicos

establecidos para el trayecto Algoritmos y Lenguajes en la resolución ME 786/09. La siguiente sección presenta el marco conceptual del trabajo. A continuación se describe brevemente la metodología de enseñanza adoptada en el las materias de Programación. Luego se vincula esta metodología con el desarrollo de la comprensión lectora, indicando los modelos de textos y lenguajes que se utilizan en estas asignaturas. Por último, se ofrecen algunas conclusiones y lineamientos para el trabajo futuro.

2 Marco Conceptual

Sobre finales de los '90 comienza en Europa la *Reforma Universitaria* a partir de una declaración conjunta que dio inicio a un *proceso de convergencia educativa*. Los principales objetivos de este proceso, que continúa vigente, son promover la cooperación para garantizar la calidad educativa, facilitar la movilidad de alumnos, docentes e investigadores y establecer un sistema internacional de créditos que permita comparar titulaciones [1]. Dada la diversidad de los sistemas educativos europeos, la expectativa no es uniformar los planes de estudio, sino adoptar un *modelo curricular basado en competencias*. El modelo busca disminuir la brecha entre la formación académica acreditada y las oportunidades para obtener un puesto de trabajo; propone establecer lo que un graduado debe *saber, saber hacer y saber ser* para poder insertarse eficazmente en el sistema productivo.

Este enfoque tiene antecedentes previos al proceso de convergencia europeo, pero es a partir de esta declaración que ha tomado relevancia en el ámbito universitario. Desde entonces el término *competencia* ha sido definido de maneras diferentes [2]. Algunos autores lo consideran equivalente a *capacidad*. Otros le asignan al término capacidad un significado estático, vinculado a una aptitud potencial, mientras que una competencia sería una capacidad puesta en acción, esto es, aplicada a una situación concreta de modo que puede evaluarse la eficacia. Bajo otro criterio el concepto de capacidad se refiere a la integración de conocimientos, aptitudes y habilidades que se adquieren en el ámbito formativo, mientras que las competencias son capacidades aplicadas al desempeño profesional.

En este trabajo definimos **competencia** como “*un conjunto de conocimientos, capacidades y actitudes que se articulan y aplican eficazmente para resolver una situación o problema concreto en un contexto académico o laboral*”. Definimos **capacidad** como una “*aptitud o habilidad que se desarrolla a partir de la integración y acumulación de aprendizajes significativos*”.

Las competencias requeridas en el sistema productivo coinciden parcialmente con las demandadas en el ámbito universitario. En las empresas esperan que los profesionales de Informática sean competitivos, proactivos, asuman riesgos, puedan liderar equipos de trabajo, enfrenten desafíos, etc. Claramente no todos los puestos de trabajo de una empresa demandan estas características, pero el crecimiento de un profesional con frecuencia está tan ligado a estas cualidades, como a su formación académica. En general los contenidos, metodologías de enseñanza y mecanismos de evaluación del plan de estudios de una carrera, no están específicamente orientados al desarrollo o valoración de estas cualidades.

Las diferencias entre las expectativas de uno y otro ámbito y la responsabilidad de la Universidad con relación al desarrollo de competencias laborales, ha sido tema de

debate en la última década. Más allá de estas diferencias, la habilidad social, el trabajo en equipo y en particular la *competencia comunicativa* son reconocidas y valoradas tanto en la etapa formativa, como durante el desempeño profesional.

Consideramos que la **competencia comunicativa** es un conjunto de conocimientos, capacidades y actitudes articulados y aplicados eficazmente para comprender y elaborar actos del habla. Un **acto del habla** es una acción que transmite un mensaje desde un emisor hacia un receptor o destinatario en un lenguaje compartido. Esta competencia requiere reconocer no solo el significado explícito o literal de un mensaje, *lo que se dice*, sino también las implicaciones, el sentido explícito o intencional, *lo que el emisor quiere decir* o *lo que el destinatario quiere entender*.

Este trabajo se refiere específicamente en la *comprensión lectora*, considerando que se trata de una de las capacidades fundamentales para la competencia comunicativa, tanto en el ámbito académico como profesional. Restringimos el análisis a las materias de Programación de las carreras ofrecidas por el DCIC.

2.1 La comprensión lectora

La lectura es un proceso de interacción entre el pensamiento y el lenguaje. Este proceso requiere reconocer los símbolos del lenguaje y las reglas que estructuran el uso de estos símbolos, pero también *comprender* el significado de lo leído. La comprensión permite elaborar un significado para el texto. El lector elabora un significado a partir de conceptos que a su vez tienen significado para él y de sus **estrategias para la comprensión lectora** [3].

Este trabajo se concentra en la comprensión de dos **tipos de textos**: *expositivos* y *descriptivos*. Estos tipos de texto aparecen en diferentes **modelos** como: *apuntes de cátedra, trabajos prácticos, libros, manuales, tutoriales, instructivos, artículos científicos, páginas web, presentaciones de diapositivas, etc.* El formato puede ser impreso o digital. Este último puede incluir recursos como sonido, animación e hipervínculos que permiten una lectura no lineal.

Cada modelo de texto puede incluir diferentes **modos discursivos**, como *definiciones, comparaciones, enumeraciones, ejemplificaciones, síntesis*, etc. Cada modelo de texto y cada modo discursivo se distingue por su *forma* y sobre todo por su *intencionalidad*.

El **nivel de comprensión** depende de la complejidad del texto y también de la formación en el tema y la *intencionalidad* del lector respecto a la profundidad de la lectura. Una *lectura primaria* o *literal* puede ser suficiente para reconocer entidades, conceptos, sucesos, acciones y relaciones explícitas. La *lectura inferencial* permite identificar causas-efectos y explicitar información o asociaciones, que están señaladas implícitamente. El reconocimiento de metáforas y analogías implica un nivel comprensión aun mayor. También se requiere un nivel de lectura comprensiva más elevado cuando una actividad exige emitir juicios o comparar entidades fundamentando la valoración. La *lectura crítica* tiene entonces un carácter evaluativo donde interviene el criterio del lector y su conocimiento acerca de lo leído [8].

2.2 La metodología de enseñanza basada en la resolución de problemas

Desde la década del '80 se han desarrollado metodologías de enseñanza basadas en la resolución de problemas con enfoques diferentes:

- Aprender *desde* la resolución de problemas implica abordar los contenidos conceptuales de una asignatura específica a partir de problemas.
- Aprender *sobre* la resolución de problemas se refiere a aprender acerca de los procesos y estrategias cognitivas que aplicamos cuando resolvemos problemas, para controlarlos en forma consciente y deliberada.
- Aprender *a* resolver problemas se orienta a que los alumnos adquieran destreza en la resolución de problemas generales, reconociendo la importancia de cada etapa del proceso de resolución.

Estos enfoques no son alternativos sino que pueden complementarse y son consistentes con la práctica reflexiva, el trabajo en equipo y por proyectos, el aprendizaje autónomo y comprometido. Demandan docentes capaces de trabajar con grupos heterogéneos, generar y gestionar situaciones de aprendizaje, entre otras competencias docentes. Las situaciones de aprendizaje deben constituir un desafío, pero al mismo tiempo deben ser una meta alcanzable [6][7].

Las metodologías de enseñanza basadas en la resolución de problemas resultan particularmente valiosas para el diseño curricular basado en competencias, dado que la resolución de problemas es una competencia fundamental tanto en el ámbito académico como laboral. Es también una metodología útil para el desarrollo de la competencia comunicativa ya que la interpretación del problema demanda capacidad para la comprensión lectora y por lo general la construcción de la solución exige habilidades para la producción escrita.

3. Las materias de Programación de las carreras del DCIC

Este trabajo analiza y describe la capacidad de comprensión lectora que se asume alcanzada y que se aspira a desarrollar en cuatro materias del área Programación: *Resolución de Problemas y Algoritmos (RPA)*, *Introducción a la Programación Orientada a Objetos (IPOO)*, *Estructuras de Datos (EdD)* y *Tecnología de Programación (TdP)*. Estas asignaturas forman parte de los planes de estudio de las carreras ofrecidas por DCIC de la UNS.

Todas adoptan una metodología de enseñanza basada en la resolución de problemas. Bajo esta concepción, el desarrollo de un *programa* puede pensarse como la construcción de una solución para un problema real, la solución es un modelo escrito en un lenguaje de programación [4].

Antes de comenzar a cursar RPA, los alumnos deben haber aprobado el curso de nivelación *Análisis y Comprensión de Problemas*, uno de cuyos principales objetivos es justamente reforzar la comprensión lectora aplicada específicamente a la resolución de problemas [5]. Durante este curso se refuerzan las estrategias para la comprensión lectora, en particular la inferencia. Se plantean problemas que permiten reconocer la importancia de interpretar adecuadamente un enunciado para identificar la incógnita, los datos explícitos e implícitos y las restricciones. Los enunciados y las soluciones incluyen textos en distintos lenguajes gráficos.

En RPA se asume que los alumnos pueden interpretar adecuadamente el enunciado de un problema como parte del proceso de resolución, son capaces de aplicar *estrategias* básicas para la comprensión lectora de distintos *modelos de textos* y reconocen diferentes *modos discursivos*. Al completar el cursado de TdP la expectativa es que

hayan reforzado las estrategias de lectura sobre un conjunto más amplio de modelos de textos, puedan comprender diferentes lenguajes artificiales y decidir el nivel de lectura que se requiere para comprender un texto, considerando la intencionalidad.

4 Lenguajes naturales y artificiales

Un **lenguaje natural** es un lenguaje con propósito general, generado espontáneamente y usado por humanos para comunicarse. Aunque existen reglas sintácticas que restringen la forma de combinar los símbolos del lenguaje, todo lenguaje natural admite cierto nivel de ambigüedad y permite que algunas expresiones tengan un significado literal diferente al sentido figurado.

En la secuencia de materias de Programación es importante que los docentes planifiquen la progresión de aprendizajes [7], consideren y decidan premeditadamente cuál es el nivel de lectura que va a requerir cada texto, qué modos discursivos y modelos de textos se proponen y diseñen actividades considerando las habilidades que asumen adquiridas para la comprensión lectora y las que aspiran profundizar, en cada etapa de la secuencia.

Un **lenguaje artificial** es una notación con un propósito específico y puede estar definido formalmente, en este caso tiene una sintaxis estricta y una semántica precisa. Como otras ciencias, la Matemática posee un lenguaje artificial que simplifica y clarifica la comunicación, designando de una manera exacta sus contenidos a través de términos y símbolos específicos. La comunicación se simplifica y clarifica si los alumnos pueden interpretar estos términos y símbolos, como así también el modo de estructurarlos. Si por el contrario, desconocen el lenguaje utilizado, la comprensión del contenido se dificulta, por más exacta y precisa que sea una expresión. De hecho, una de las dificultades de los alumnos que ingresan a una carrera universitaria se relaciona justamente con la interpretaciones de textos escritos en una *notación formal* como la que brinda la Matemática.

Las materias de Programación ofrecen múltiples oportunidades de aplicar notaciones formales, en particular el lenguaje Matemático aprendido en el Nivel Polimodal y profundizado en las asignaturas de Ciencias Básicas. Es importante que se reflexione explícitamente acerca de los beneficios de utilizar una notación rigurosa y se generen actividades que incluyan entre sus objetivos la “lectura” de textos escritos usando esta notación. Una especificación como *Sea S una secuencia de la forma $s_1 s_2 \dots s_{m-1} s_m$ tal que $s_i \leq s_{i+1}$ con $1 \leq i < m$.* va a requerir que el docente describa la notación al comenzar el cursado de RPA y ayude a interpretarla tanto en las clases teóricas como prácticas, manteniendo por supuesto la uniformidad en el lenguaje. Al completar esta asignatura la lectura de este tipo de expresiones debería ser fluida.

Así, la *semántica de las operaciones* que los alumnos deben implementar comienza a describirse combinando lenguaje natural y una notación formal desde la primera materia de Programación. La definición recursiva de factorial, el algoritmo para computar el máximo común divisor entre dos números naturales, las propiedades de secuencias de elementos, la especificación formal de las operaciones del tipo de dato abstracto Pila, son algunos de los ejemplos con los cuáles se aprenden a usar notaciones formales para establecer la semántica de las operaciones.

La interpretación de cada una de estas especificaciones se puede complementar con algunos ejemplos significativos, que luego pueden ser usados como casos de prueba en la verificación. Es importante que los alumnos entiendan que la solución tiene que ser general, no solo funcionar correctamente para los casos de prueba específicos.

4.1 Lenguaje natural

En las cuatro materias de Programación en las clases teóricas se utilizan presentaciones de diapositivas. Cada docente aplica diferentes criterios y estilos para armar sus presentaciones y combinarlas con otros recursos, como por ejemplo el pizarrón, el ambiente de programación o animaciones para ilustrar la ejecución de algunos programas. Cualquiera sea el criterio y estilo, las presentaciones siempre incluyen y destacan el *vocabulario técnico* que se va introduciendo junto con los contenidos conceptuales propios de cada materia. Así, el lenguaje natural conocido por los alumnos, aumenta con nuevas palabras, propias de la disciplina.

Aunque asumimos que los alumnos pueden reconocer diferentes *modos discursivos*, en general resulta valioso detenerse para resaltar las características de cada uno de ellos con respecto a la *forma* y sobre todo la *intencionalidad*. Asimismo es pertinente remarcar antes de una evaluación, que cuando se pida la definición de un concepto, no se considerará válida una respuesta que sólo aporte algunos ejemplos, aunque se refieran a ese concepto; del mismo modo, si una consigna requiere una comparación entre dos recursos, es probable que dos descripciones independientes no se consideren válidas como respuesta, excepto que ambas describan a los conceptos en términos de los mismos atributos y la comparación sea evidente, aunque no esté explícita.

La lectura de manuales y tutoriales es una actividad en la cual los alumnos tienen que ir adquiriendo destreza progresivamente. Es importante que se incluyan actividades que requieran interpretar textos de temas no desarrollados en clase, pero que al mismo tiempo tengan una complejidad adecuada para ser comprendidos de manera autónoma. Cuando los manuales o tutoriales están disponibles en versión digital es posible estimular la lectura no lineal proponiendo actividades que requieran navegar a través de hipervínculos.

En RPA los contenidos conceptuales y el lenguaje de programación utilizado se presentan en detalle en las clases teóricas a través de la resolución de problemas. El lenguaje natural aparece entonces en las descripciones, definiciones, comparaciones y enunciados. Los enunciados incluidos en trabajos prácticos y evaluaciones, con frecuencia se complementan con textos escritos en lenguajes artificiales, por ejemplo algoritmos, expresiones matemáticas, diagramas.

Es importante que el docente reflexione cuando produce una actividad, cuál va a ser nivel de lectura que demanda. Más allá de la dificultad que pueda entrañar el problema, si los datos, la incógnita y las restricciones están formulados explícitamente, la comprensión de texto no demandará mayor esfuerzo. Si en cambio la interpretación del enunciado requiere un proceso de abstracción para reconocer los datos relevantes o un proceso de inferencia que permita identificar datos que están implícitos, la capacidad de comprensión toma un rol más importante para la resolución.

En las dos primeras materias de Programación, como así también en el curso de nivelación, se incluyen problemas equivalentes en complejidad respecto a las estrategias de resolución, pero con diferentes requerimientos respecto al nivel de

comprensión lectora. La expectativa es justamente identificar las dificultades para poder superarlas.

En IPOO los elementos básicos del lenguaje de programación se describen brevemente en las clases prácticas y se asume cierta autonomía por parte de los alumnos para profundizar los temas presentados y evacuar dudas consultando manuales, tutoriales y por supuesto a los docentes. Algunos paquetes y clases provistas por el lenguaje se describen parcialmente y se propone como actividad investigar acerca de otros los servicios. En ningún momento la expectativa es que los alumnos los memoricen. Los recursos provistos por el lenguaje para soportar los conceptos centrales de la programación orientada a objetos, se presentan en teoría, se aplican en la práctica y se refuerzan a través de la lectura de bibliografía.

En EdD y TdP la lectura profunda de la bibliografía propuesta es indispensable para las evaluaciones finales y la realización de proyectos. Una de las cuestiones que afecta a la adaptación de los alumnos al ámbito universitario es justamente el nivel de profundidad con el que leen y nivel de lectura con que se espera que lean. Una lectura superficial de un texto puede ser muy adecuada entre una clase y otra, de modo que el docente pueda profundizar en un tema o incluso presentar uno nuevo, asumiendo que los alumnos comprendieron las nociones básicas del contenido presentado previamente. El mismo nivel de lectura no va a ser adecuado cuando se espera que comprendan en profundidad un tema, por ejemplo para aplicar los contenidos para la resolución de un problema o la realización de un proyecto.

En cada materia es importante estimular la reflexión acerca de cuándo es suficiente realizar una lectura *primaria* y cuándo se espera que alcancen un nivel de comprensión más profundo a través de una lectura *inferencial* o más aun, *crítica*. Un aspecto a remarcar es que la lectura de un resumen de un texto puede ser más compleja que la del texto original, porque en el resumen suelen quedar implícitas asociaciones y se eliminan aclaraciones, observaciones, analogías y ejemplos que favorecen la comprensión. Este hecho no siempre es evidente para los alumnos y con ejemplos adecuados puede ilustrarse claramente. Este tipo de actividades, que en principio no son parte de los objetivos de estas asignaturas, pueden resultar fundamentales para favorecer capacidades como la comprensión lectora.

En cada una de las tres primeras materias de Programación se utiliza un entorno de desarrollo diferente. Desde el punto de vista de la comprensión lectora un ambiente de desarrollo demanda interpretar las opciones provistas por los distintos tipos de menús, los mensajes de error del compilador y la información que nos brinda el depurador. Actualmente los alumnos están habituados a interactuar con diferentes interfaces gráficas de usuarios y en general se adaptan rápidamente a explorar y utilizar una aplicación de software. También es común que la instalación de una aplicación no implique una dificultad. Aun así, es conveniente proponer alguna actividad que implique leer un instructivo como por ejemplo para instalar un entorno de desarrollo, crear un paquete o un ejecutable.

4.2 Lenguajes Artificiales

Uno de los objetivos de las materias de Programación es que los alumnos interpreten textos escritos y construyan soluciones usando diferentes lenguajes. Algunos de estos lenguajes tienen una especificación formal como los *lenguajes de programación*, *lenguajes de modelado*, *lenguajes gráficos de especificación sintáctica*, *notaciones*

matemáticas, etc. Otros lenguajes artificiales no tienen reglas formales que los definan como por ejemplo los *lenguajes de diseño de algoritmos* y los *lenguajes para modelar la evolución de la memoria*.

En cada materia es necesario planificar la presentación de los lenguajes en las clases teóricas respecto a cómo se utilizan y combinan en los enunciados de problemas y proyectos en las clases prácticas y en las evaluaciones parciales y globales. Un objetivo más importante aun, es que los alumnos *aprendan a aprender* lenguajes artificiales. El enfoque basado en resolución de problemas es fundamental en este sentido porque cada lenguaje se aprende *usándolo*. Así, más allá de describir los aspectos sintácticos y semánticos, es necesario centrarse especialmente de las cuestiones pragmáticas de cada notación. El uso de distintas notaciones artificiales contribuye a desarrollar la *autonomía en el aprendizaje*. Es muy importante articular la secuencia de acuerdo a la cual se presentan y utilizan cada una de estas notaciones en cada asignatura, para que los alumnos puedan desarrollar gradualmente la capacidad de comprender y producir textos escritos utilizando distintos lenguajes.

En RPA se propone un **lenguaje de diseño de algoritmos** que, con una sintaxis menos rigurosa que el lenguaje de programación, favorece la construcción de un modelo para la resolución de cada problema planteado. El objetivo de esta etapa es identificar las estructuras de control requeridas para la resolución. El lenguaje de diseño resulta útil cuando el problema puede ser dividido en subproblemas o la resolución implica combinar al menos dos estructuras de control. Si el problema es muy simple, no se destaca la ventaja de diseñar un algoritmo antes de comenzar a escribir código en el lenguaje de programación.

También en RPA se introduce un **lenguaje para modelar la evolución de memoria**, que permite construir diagramas que muestran una traza de la ejecución de un programa o subprograma. Aunque existen diferentes formas de construir estos diagramas, es importante acordar una notación uniforme dentro de una misma asignatura. En IPOO se usan **diagramas de evolución de referencias** que modelan la memoria, pero en este caso en el paradigma orientado a objetos.

Los **lenguajes gráficos de especificación sintáctica** se utilizan en las cuatro materias descriptas en este trabajo. La sintaxis de cada mecanismo provisto por los lenguajes de programación usados, se especifica a través de *diagramas sintácticos*. En RPA la aparición de un diagrama requiere detenerse en explicar la notación, más allá del objeto mismo que se está describiendo. En general no se destina una clase específica a describirla, sino que se presenta en forma transversal a otros contenidos. Es muy importante articular las materias y acordar en qué momento se espera que los alumnos tengan autonomía para la lectura de este tipo de diagrama.

En IPOO los alumnos interpretan diagramas sintácticos, reconocimiento especificaciones condicionales, iterativas y recursivas. La “lectura” de un diagrama sintáctico permite reconocer instrucciones y expresiones válidas, distinguir aspectos sintácticos y semánticos de un lenguaje de programación y comprender la necesidad de que un lenguaje de programación tenga una sintaxis rigurosa.

Los diagramas sintácticos se utilizan también para especificar la forma de una secuencia de elementos que deben ser procesados. La resolución del problema requiere interpretar la especificación y luego diseñar el algoritmo para procesar la secuencia e implementarlo en un lenguaje de programación. Si los alumnos están

familiarizados con este tipo de diagrama, la formalización es una ayuda porque permite descubrir las estructuras de control adecuadas y facilita el diseño del algoritmo. En caso contrario, la “lectura” del diagrama puede representar una dificultad adicional al problema propiamente dicho.

En IPOO se introducen los elementos básicos de un **lenguaje de modelado** para construir *diagramas de clases* y en las materias que siguen se presentan otro tipo de recursos para elaborar distintos tipos de diagramas. En IPOO muchos de los enunciados incluyen un diagrama de clases en donde se especifican las decisiones de diseño de cada clase y las relaciones que las vinculan. La interpretación del diagrama, incluyendo las responsabilidades de cada clase, es una de las capacidades que se aspira desarrollar en esta asignatura. En las materias que siguen no solo se interpretan modelos escritos en UML sino que muchos problemas requieren producir diseños usando este lenguaje.

La comprensión lectora de textos producidos usando los lenguajes artificiales mencionados, es una capacidad que las materias de las áreas Ingeniería de Software y Sistemas, asumen adquiridas previamente. El desarrollo de esta capacidad se alcanza como tanto en las materias de Programación como en las asignaturas del área Fundamentos de Ciencias de la Computación.

Conclusiones y trabajo futuro

Las competencias comunicativas son reconocidas y valoradas tanto en el ámbito académico como en el sistema productivo. Los planes de estudio de las carreras ofrecidas por el DCIC de la UNS no incluyen asignaturas orientadas específicamente al desarrollo de la comprensión lectora, la producción escrita y la oralidad. Estas capacidades se consideran transversales, de modo que el diseño curricular de las asignaturas se realiza asumiendo cierto nivel adquirido de cada una de ellas y se compromete a reforzarlas.

En las materias del área Programación de estas carreras se adoptó hace más de 20 años una metodología de enseñanza basada en la resolución de problemas. El objetivo principal de aprendizaje está vinculado evidentemente con el desarrollo de programas, para lo cual se presentan varios lenguajes de programación.

La comprensión del lenguaje natural se enriquece con la lectura de diferentes *modelos de textos*. Es importante que los docentes hagan notar a los alumnos las particularidades de cada modelo, especialmente en lo que se refiere a la comunicación, e indiquen también el nivel de lectura que se espera que realicen de cada uno, en cada momento. La caracterización de tres *niveles de lectura* puede ser suficiente en este sentido: primaria, inferencial y crítica.

Cualquiera sea el modelo de texto puede incluir diferentes *modos discursivos*. Estos modos son conocidos previamente por los alumnos pero la reflexión explícita acerca de las características y sobre todo la intencionalidad de cada uno, contribuye a fortalecer las competencias comunicativas. En particular, las definiciones aumentan el vocabulario incorporando términos propios de la disciplina y enriqueciendo el lenguaje natural.

La introducción progresiva de diferentes *lenguajes artificiales*, con distintos grados de rigurosidad sintáctica y semántica, contribuye a desarrollar la competencia

comunicativa. Es importante articular adecuadamente la presentación y aplicación de cada nueva notación, de modo que cada una constituya una ayuda y no un obstáculo. Algunos lenguajes van a ser descriptos en detalle, siempre aplicados a situaciones concretas, en otros se va a requerir mayor autonomía de aprendizaje por parte de los alumnos. En cada caso debe reflexionarse y establecerse explícitamente las competencias previas que se asumen adquiridas.

Este trabajo se concentra específicamente en la comprensión lectora abordada desde cuatro asignaturas de una misma área. Una extensión natural es considerar la producción escrita y la oralidad. El trabajo puede extenderse también para abordar el desarrollo de la competencia comunicativa en otras áreas de la disciplina. Una vez que se haya avanzado en estos dos sentidos la expectativa es analizar otras competencias reconocidas tanto en el sistema educativo como productivo.

Referencias

- [1] Declaración de Bolonia. Declaración conjunta de los Ministros Europeos de Educación. Bolonia (1999) <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/>
- [2] Mastache, Anahí: *Formar personas competentes. Desarrollo de competencias tecnológicas y psicosociales*. Novedades Educativas. ISBN:978-987-538-199-5 (2007).
- [3] Castro Fox Guillermina, González Nora, Monti Carla, Palmucci Daniela: *Estrategias para la Lectura Comprensiva. Un abordaje al discurso científico-pedagógico*. Universidad Nacional del Sur. (2005)
- [4] Rueda Sonia, García Alejandro: *Análisis y Comprensión de Problemas: Fundamentos, Problemas Resueltos y Problemas Propuestos*. Notas del curso de nivelación. Universidad Nacional del Sur. Argentina Págs.: 90. (2004)
- [5] Rueda Sonia, Señas Perla: *Un enfoque basado en la resolución de problemas para la enseñanza de la Programación Orientada a Objetos* Bahía Blanca III Congreso en Tecnología en Educación & Educación en Tecnología (Teyet) (2008)
- [6] Perrenoud Philippe: *Diez nuevas competencias para enseñar*. Editorial Graó. ISBN 978-84-7827-321-8 (2004)
- [7] Whimbey Arthur, Lochhead Jack, Narode Ronald: *Problem Solving & Comprehension* Wearset Ltd. Boldon ISBN 978-0-415-50221-4 (2013)
- [8] Alonso Tapia, Jesús: *Leer, comprender y pensar: Desarrollo de estrategias y técnicas de evaluación*. MEyC. CIDE. Madrid. ISBN: 84-369-2270-0 (1992)